# Varying Accuracy Configurable Multipliers integrated by Utilizing Quality 4:2 Compressors

**S Aruna, M. Dheeraj, K. Srinivasa Naik**

***Abstract— In this paper, few 4:2 compressors which are having the flexibility to switch between two operating modes namely exact and approximate based on the requirement of the application, along with an exact compressor that has greater characteristics that the conventional compressor are put-forward. The approx. approach enables these twin-quality compressors to deliver high-speeds with low power consumption at the cost of reduced accuracy which enables them to be used in various solicitations of their importance. Each of these compressors has their own levels of accuracy with different delay, power dissipation and areal consumption values in the approximate mode as well as in the exact mode too. Usage of these configured compressors in the structure of the multipliers helps in further optimizing the properties of designing of multipliers. The efficiencies of these 4:2 compressors are evaluated in 8-bit Dadda multiplier in the 45nm standard CMOS technology by relating their parameters with those of the up-to-the-minute approximate Multipliers. Using a calling circuitry 16 & 32 (bit) dadda multipliers are also evaluated incorporating them with the proposed compressors. This comparative evaluation results indicate reduced delay and lowered power consumption at the cost of reduced accuracy in the inexact mode of the twin eminence compressors, whereas the EXACT compressor that has been designed showcases lowered power, improved speed and reduced area on silicon with precise accuracy in the results. Also, the effectiveness of the EXACT compressor is used to optimize a MAC unit which is used vastly for many solicitations.***

***Index Terms— Approximate operating mode-computing, 4:2Compressor, Accuracy, configuring, Delay(lag), Power.***

## I. INTRODUCTION

Integration (LSI, VLSI, and GSI) refers to devices assembled at very high densities resulting in advanced circuitry that can perform crucial operations of very high order. It has certain characteristics and constraints which are to be optimized in all its essential components for their designing and implementation such as Speed, Area on silicon, Power and Accuracy. Considering these optimization entities, any block that is highly used in VLSI design architecture must be optimized so that it should be advantageous and must follow these constraints.

Most electronic, Arithmetic solicitations are implemented using digital logic circuits. Summation and multiplication are extensively employed maneuvers in electronic and computer arithmetic, which be there employed by their respective blocks. Multiplication of data stays a mathematical manner Maneuver that at its modest is an abridged route of totaling an integer to itself for definite repetitions. A no. which is the multiplicand is added to itself a no. of times as quantified by another no., the multiplier to form a result. A fast Multiplication course consists of three steps: Partial Product Generation, Partial Product Reduction and Final stage-Addition. Amongst the various arithmetic components, the multiplier is one of the crucial components, which is extensively used in different solicitations such as multimedia handling and its tenders, Microprocessors, Microcontrollers…etc. According to the statistics more than 70% instructions in the microprocessors and most of the DSP algorithms perform addition and multiplication. Hence the demand of high speed and low power processing is highly necessary with the increasing need of Multipliers in operation. One of the techniques is, by reducing the number of operations, the switching speed can be increased and the power consumption can be lowered to a notable extent.

Multipliers are of many types based on their simplification and partial product gaining techniques such as Serial Multipliers, Parallel Multipliers, Bit Array Multipliers, and Column Compression Multipliers. Among these, the Column Compression Multipliers are highly utilized due to their maximum optimizing natures which are further under development. The usual process of multiplication of binary data is done as shown below in the Figure 1 (a) and its dot structure is shown in Figure 1 (b).

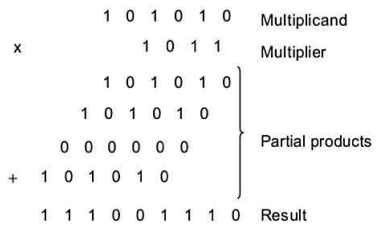


**Figure 1 (a): Multiplication process**

**Revised Manuscript Received on February 11, 2019.**

**Dr. S Aruna,** Department of ECE, Andhra University College of Engineering (A), Visakhapatnam, India (Email: aruna9490564519@gmail.com)

**M. Dheeraj,** Department of ECE, Andhra University College of Engineering (A), Visakhapatnam, India (Email: dheeraj.chess@gmail.com)

**Dr. K. Srinivasa Naik,** Associate Professor, ECE, Vignan's Institute of Information Technology(A), Visakhapatnam, India (Email: nivas97033205@gmail.com)

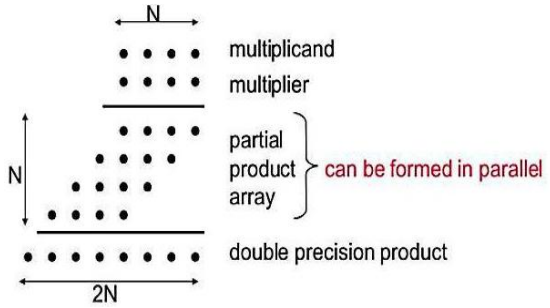


**Figure 1 (b): Corresponding Dot structure**

The main components of a multiplier are Multiplicand and Multiplier which results in partial products and finally summed up to the result. Always the optimization of a single factor is done bearing the other factor as a constraint. Unambiguously, resulting in improved performance characteristics limiting the area and power factors of the systems is of greater challenge. In addition, another obstacle that can be considered is to reach the required level of reliability to sustain the system performance.

For meeting the design specifications such as limited power, improved speed and all various methods of designing have been suggested at different levels of abstraction. One of the highly used designing methods in computing is the Approximate computing. Approximate calculating can shrink the design intricacy with a rise in performance and power efficacy for fault resistant solicitations. In solicitations like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts. Research on approximate computing for error tolerant solicitations is on the rise. Adders and multipliers form the key components in these solicitations. Approximate computing techniques mainly target on accomplishing the required provisions at the cost of reduced computation precision.

These approaches possibly will be castoff in solicitations where there is not an exceptional reply and/or a set of exceptional outcomes near the precise product that can be well-thought-out acceptable i.e. as the name suggests it is the computation of problems approximately rather than deterministically which leads to an acceptable loss of accuracy and saves on other important parameters like circuit area, power, delay timing. These solicitations consist of multimedia handling, Machine learning, Signal Processing, and various other error resilient computation used application. Approximate arithmetic units are mainly based on the simplification of the arithmetic used circuitry. There are many such mechanisms that are converging on approximate multipliers, which enable sophisticated speeds also reduced power feedings at the rate of varied accuracies. More or less, whole of the offered approximate multipliers be there centered on requiring a static level of precision throughout the run-time. The accuracy at runtime re-configurability, nevertheless, is deliberated as an essential characteristic for giving those altered levels of worth and services during the task of the system.

At this juncture, by tumbling the uniqueness of results, the interruption and power consumption of the system possibly be condensed. In addition to this, few digital units, for instance broad purpose processors, could be utilized for both the nominal & the precise calculation techniques. One of the greater methodologies for accomplishing this specification is to use an approximate entity sideways by a respective rectification entity whenever necessary. The rectification entity, nevertheless, brings out the previous existing results such as increasing the lag, power, and SOC overheads of the equipment. Furthermore, this error rectification route would entail additional clock cycles, which might, in chance, further slowdown the handling procedure. Among the various types of multipliers, Column compression multipliers are highly utilized due to their advantages of satisfying VLSI constraints. High performance circuits such as CPA (Carry Propagation Adders), Bit Array Multiplier, which are non-column compression type and Wallace tree Multiplier, are highly used but had reached a stage of saturation in terms of optimization resulting in search of alternative optimization circuitry such as Dadda multiplier. The Dadda Multiplier is a hardware multiplier conceived via a computer technologist “Luigi Dadda” in 1965. It is analogous to the Wallace multiplier, and then it is faintly rapid (for any operand sizes) and involves rarer gates (for all but the smallest operand sizes). In fact, Dadda, Wallace multipliers ensure the same 3 steps for 2 bit strings w1 & w2 of dimensions l1 and l2 individually:

1. Multiply (logical AND) each bit of w1, to each bit of w2, yielding l1.l2 results, assembled by weight in columns.
2. Condense the no. of fractional harvests by phases of full and half adders unless we are remained with max 2 bits of every single weight.
3. Add the ultimate end result thru a conventional adder.

For example, the product (×) of bits anbm has weight n + m. Unlike Wallace multipliers that reduce as much as possible on each layer, Dadda multipliers shot to diminish the number of gates used, as well as input/output delay. Because of this, Dadda multipliers have an inexpensive declination segment, but the ultimate no. s could be a few bits lengthier, thus calling for marginally superior adders. It has a similar structure of Wallace tree but mainly differs in the usage of adders as Dadda uses as few adders as possible by maintaining minimum number of reductions.

Because of this the reduction stage of Dadda multiplier is inexpensive further reducing its delay. But the issue is that the rules of Dadda reduction are a bit complex when compared to others due to its structural intake.

The key advantage of Dadda multiplier is that some of the 3:2 compressors / Full adders in the reduction flow can be replaced with 4:2 and other kinds of Compressors easily.

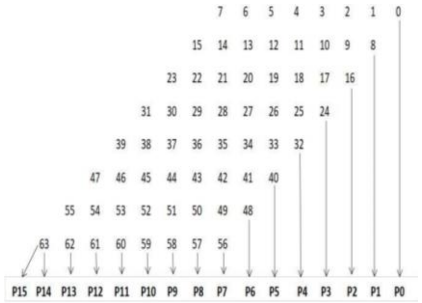


**Figure 1(c): Dadda Multiplier Reduction Partial product matrix for 8x8 Multiplier: Type-1**

This Figure 1(c) is the basic partial product matrix to simplify a Dadda multiplier which is already optimized but couldn't help to achieve the desired characteristics to the heights of the expectation. Figure 1(d) is the updated one.

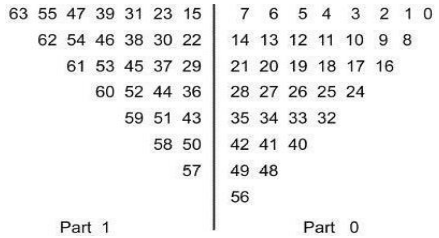


**Figure 1(d): Partial product matrix after Rearranging, Reorganizing and Splitting into two parts: Type-2**

## II. STATE OF ART

As already plenty of exploration in crafting the nominal (approx.) multipliers are present, the exploration struggles on accuracy variable nominal multipliers are inadequate as well as the effort to design an EXACT accuracy multiplier with varied circuitry that supports the design metrics didn't take place. Here we review some of these previously done works. As per broadcast [10], an SS method (Static Segment) has been given out. It executes the multiplying act on an *m*-bit sector which starts as of the prominent 1-bit of the in-feed species having m >= n/2. Therefore, an ($m \times m$) multiplier occupies lesser energy > than an ($n \times n$) multiplier. A DRUM-Dynamic Range Un-biased Multiplier takes an *m-bit* sector. It starts from the prioritized 1st-bit of the in-feed specimens, and makes the LSB (Least Significant Bit) of the truncated operands as 1. It was the same as given out in the paper [11]. As per the given skeleton, the truncated outcomes are multiplied. They are further forwarded to left side harvesting the final values. Although, by playing on minute values for *m*, its outline of it shown in [11] gives us superior & accurate designs rather than which are shown in [10] would provide. This tactic facilitates highly complex circuitry to proceed further to the result.

A broken array multiplier is a bioinspired approximate device. It was depicted in the paper [13]. In the structure that is proposed, nearly few CSA cells are used in both rows and columns all through the addition of the intermediate products. This has been mislaid to protect the consumptive-power & minimize the SOC & lag (delay) factors. In the paper [14], 2 approximate 4:2compressors are depicted and equipped in a build of the Dadda-multiplier. These advised compressors always worked in a single style which is the approximate mode. In the paper [1] the K map of a (2 × 2) multiplier (removing 1-term from its K-map) is modified. A nominal (2 × 2) multiplier thru a meeker structure was been given out here. This structured slab that is been proposed is useful for fabricating bigger sized multiplier structures also. Also, here a blunder recognition and rectification circuit has been proposed. An erroneous multiplier proposal of restructuring the multiplier into 2 multiplication and non-multiplication segments stayed hosted in [12]. The product calculating slot was erected depending on the theoretical multipliers while the non-multiplication portion was instigated in a nominal edifice with an indicated rate of blunder. Both the proposals in [12] and [1] suffers high relative blunders. In the paper [15], a mighty precision approx.4×4Wallace-Tree multiplier was entitled. It incorporated a 4:2approximate poker chip which leads to the delay & power diminutions of the fractional product (×) stages of the 4×4Wallace. In this broadside, the anticipated smaller sized multiplier was castoff to form superior multipliers. Due to the structure of array, this nominal multipliers delay was outsized. Besides, an ED was advised to be castoff at the o/p of the nominal 4 × 4Wallace tree. The entity bred the precise harvest in the instance of the Exact-functioning manner. As in-[16], by propositioning a nominal summer by means of a tiny carry proliferation lag, the fractional product shrinking stages were flown up. In this paper, an OR-gate-based error reduction unit has also been proposed. In the paper [17], a ROBA was anticipated that rounds the in-feed variables into their closest exponent of 2. It must be perceived that the blunder rescue entity (those in [1], [12], [15], and [16]) upsurges the consumed-power & delay of the multiplier. This entails that the precision refrangible multipliers would ensure larger delay & power overheads.

## III. WHY COMPRESSORS

In order to diminish the time-taking of the intermediate products of the adder stages of multipliers, for providing high speed and lower power consumption with minimized area, Compressors are equipped instead of regular Adders. Compressors based on their sizes can minimize as many inputs at a time resulting in improved speed, reduced delay, minimized area on chip and lowered power consumption. The basic adders- Half adder and full adder can only minimize two, three inputs at a time where they are said to be 2:2 compressor and 3:2 compressor respectively Figure 3(a), Figure 3(b).

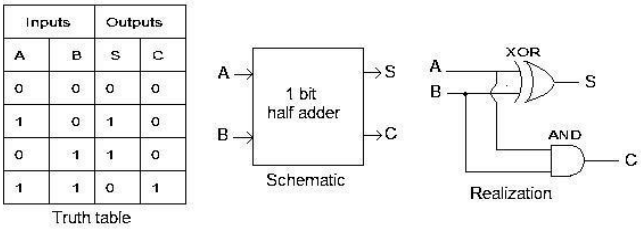


**Figure 3(a): Half-adder also termed as 2:2 Compressor**

| a | b | $C_{in}$ | sum | carry |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

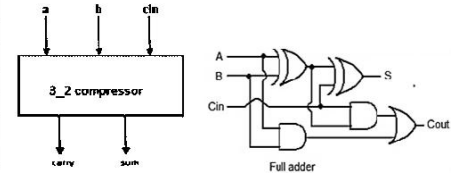


**Figure 3(b): Full-adder also termed as 3:2 Compressor**

In this paper, few 4:2 compressors are proposed, that can ensure occurrence of switching amongst the exact and approximate manners with a very minute delay & power outflows by incorporating a supplementary part to the approximate one. These could support a wide range of solicitations with varying accuracy need.

## IV. PROPOSED 4:2 COMPRESSORS

Hereby few 4:2 compressors are proposed. A 4:2conventional compressor minimizes four inputs into two along with an input carry bit.

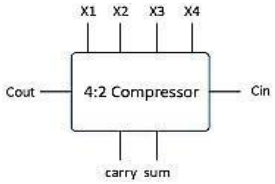


**Figure 4(a): Block illustration of a 4:2compressor.**

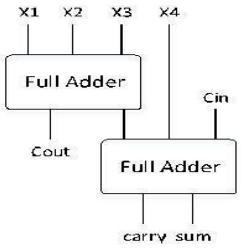


**Figure 4(b): Structure of a conventional 4:2 compressor**

Considering the 4:2 compressors to be equipped in their necessary situations, we propose four Dual quality Approximate compressors that yield an approximate level of accuracy in the results after the execution. These all have their own percentages of accuracy in the results enabling their usage in their respective useful solicitations.

The compressors being proposed performs in 2 variant operating styles i.e. Approximate and Exact. A general view of these compressors is shown through a block diagram in Figure 4(c).

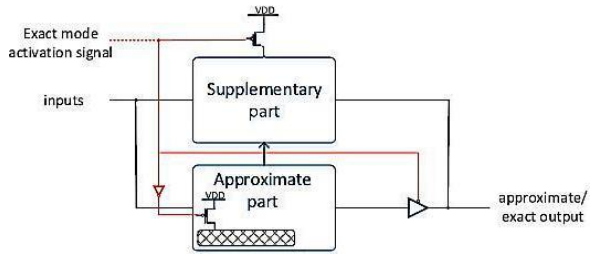


**Figure 4(c). Block illustration of the offered 4:2compressors mode of operations.**

The block illustration consists of two parts i.e. the Approximate and the Supplementary part. The proposed compressors work in the approximate manner utilizing that part alone. If in case of necessity to improve the accuracy the supplementary part can be attached to the approximate part and be executed to yield results. These designs have different accuracies, power, delay and areal consumption due to their different structures of designing both in the approximate as well as in exact modes of operation.

*Structure-1 (DQ4:2C1):*

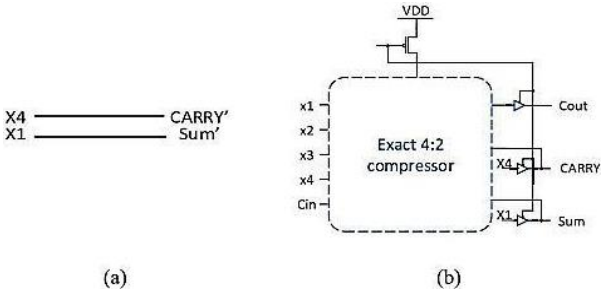


**Figure 4 (c): Nominal portion & (d) Whole build of DQ 4:2C1**

For the 4(c) part of the given out DQ4:2C1 edifice, as depicted in Figure4(c), the nominal resultant "carry" (→*carry*) is been directed to *x*4 (*carry*= *x*4), & in the same way, its *sum* is directed to *x*1 (*sum*). In the (a) portion of this build, the harvest *C*out is overlooked. But the nominal portion of the DQ4:2C1 structure is considerably faster and the lower powered. Its blunder rate is high till 35%.

The supplementary (b) part of this structure results in an exact 4:2 compressor itself. The whole assembly of this varied compressor is as revealed in Figure 4(d). In the EXACT phase, its lag (delay) is almost same as to that of the conventional structure of the 4:2compressor.

*Structure-2 (DQ4:2C2):*

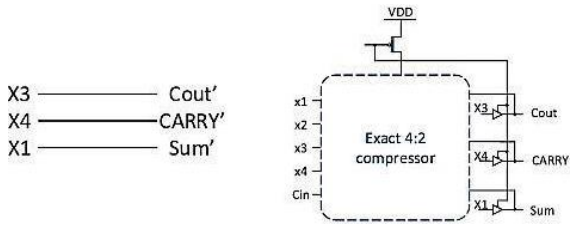


**Figure 4 (e): Nominal portion & (f) Whole build of DQ4:2C2**

In the first DQ4:2C1 structure, the Cout is ignored and the internal structure has been simplified and the corresponding error rate is lowered. In the 2nd depiction, compared with the 1st, the *C*out is spawned by directing it to the *x*3 input in the (e) part of Figure 4(e), The Figure 4(f) illustrations the core edifice of the approximate block and the whole edifice of DQ4:2C2. While the blunder rate of this edifice is identical to that of DQ4:2C1, which is 35%, its relative error is lower when compared to the other.

*Structure-3 (DQ 4:2C3):*

The former edifices, in the nominal (approx.) functioning phase, had 'power-delay' drops to their max extent when related with the EXACT compressor. In some solicitations, conversely, the precision need is high. In the 3rd edifice, the exactitude of the nominal work phase is upgraded by aggregating the intricacy of the unprecise slot which has its core edifice as depicted in Figure 4(g). In this edifice, the precision of harvest i.e. *sum* is amplified. Analogous to the first proposed one, the precise slot of this edifice won't back the resultant-*C*out. The blunder degree of this build fell to 25%. The whole build of DQ4:2C3 is depicted in Figure 4(h).

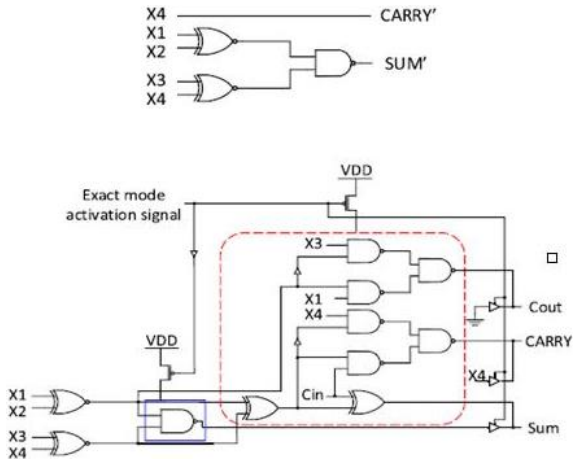


**Figure 4 (g): Nominal portion and (h) Whole block of DQ4:2C3**

A red-dashed line made rectangle is used to depict the supplementary block. The NAND-gate that is intricately showcased in a blue dotted-lined rectangle won't be used in the EXACT phase. Hereafter, while using this functional phase removal of it is highly suggested by any vital technique.

*Structure-4 (DQ4:2C4):*

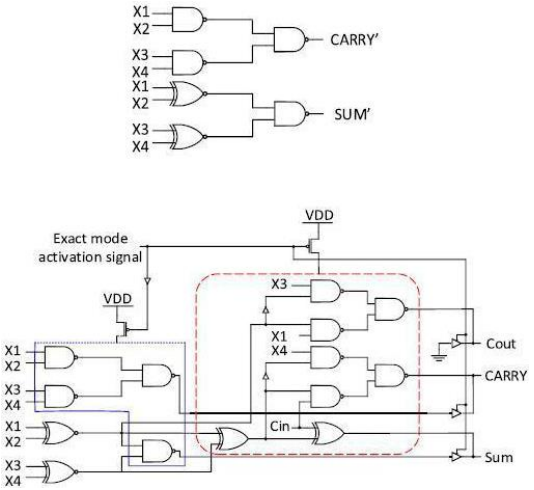


**Figure 4 (i): Nominal portion and (j) Whole portion of DQ4:2C4**

Development of the precision of *carry* related to DQ4:2C3 at the price of huge delay & more power feedings wherever the blunder rate is diminished to 20% is done.

The core build of the nominal portion & also the whole edifice of DQ 4:2C4 are exposed in Figure 4(i), 4(j). The supplementary portion is specified thru red-dashed-rectangle. The gates of the nominal portion are turned-OFF throughout the EXACT functioning phase. These exist depicted by some blue-dashed marks.

*Note*: The blunder level resembles to the happening of the no. of blunders in the harvest for whole series of the inputs.

*Exact 4:2 Compressor:*

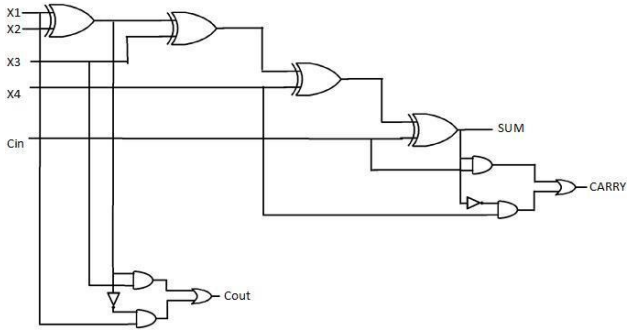


**Figure 4 (k): An EXACT compressor**

To support the reduction of the lag of the fractional products of the summing stages of the parallel-multiplier, 4:2compressors are broadly engaged. The effort of this paper is on approximate 4:2compressors, EXACT-compressor with reformed factors. Mainly, some upbringing on the exact 4:2compressor is obtainable as it has the desired features.

These sort of compressor shown schematically in the Figure 4(k), have 4 inputs ($x1$–$x4$) along with a carry input (*C*in), and two outputs (*sum* & *carry*) sideways with a *C*out out-put. The core build of an exact-4:2compressor consisted of 2 in sequence directed full-summers as known. In the build, the masses of all the input-feeds and the *sum* output-feed are equated. But the masses of the *carry* and *C*out harvests are 1-binary bit level peak. The harvests *sum*, *carry*, and *C*out do obtain commencing:

*Boolean expressions of EXACT compressor:*

$$\text{sum} = x1 \oplus x2 \oplus x3 \oplus x4 \oplus Cin$$
$$\text{carry} = (x1 \oplus x2 \oplus x3 \oplus x4)Cin + \overline{(x1 \oplus x2 \oplus x3 \oplus x4)}x4$$
$$\text{Cout} = (x1 \oplus x2)x3 + \overline{(x1 \oplus x2)}x1.$$

*The Multiplier Architecture*

There are two kinds of depictions of partial product matrix for the column compression type multipliers Figure 2(a) and Figure 2(b). Here the Type-2 one which is the reorganized matrix is of utmost priority due to its efficiency in yield. If we consider the 4:2 compressors as proposed, the usage of them in Dadda multiplier is as shown in the below Reduction diagram Figure 4(l).

The Dadda multiplier circuit can equip many kinds of compressors in it and help out to reduce the power

consumption, delay and area which make its usage more than other multipliers for optimization. This application of dadda multiplier is highly useful for larger circuits in various solicitations such as Multimedia processing, Machine learning.

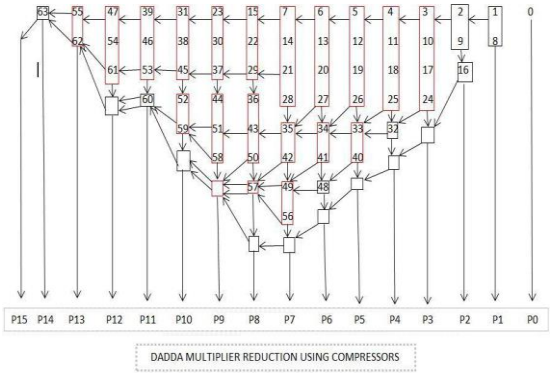


**Figure 4(l). Dadda Multiplier reduction using compressors**

The blocks represented in the diagram with numbering-1,2,3,.. are the terms to be minimized such as a0b0, a0b1, a0b2,…,anbn, where it is a8b8 in the case of 8-bit multiplier.

## V. CONVERSION CIRCUITRY

*16-bit multiplier generated using 8-bit multipliers:*

Equipping the proposed compressors, an 8-bit multiplier has been designed and their respective outcomes have been understood. In order to design 16-bit multiplier and 32-bit multiplier, a calling circuitry has been used so that the previously designed 8-bit multiplier can be used according to the necessity and the result can be achieved.

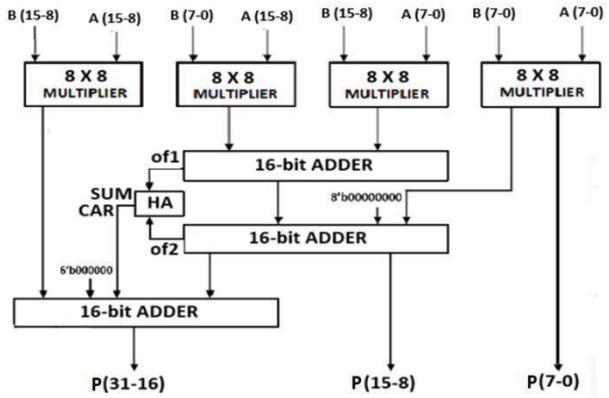


**Figure 5(a): 16-bit multiplier design using 8-bit multipliers**

Here the 16-bit ADDER block represents a regular concatenation adder to sum up the given two 16-bit inputs which are the outputs of the previous stage of the 8-bit multipliers. The A (0-7), A (8-15) and B (0-7), B (8-15) are the binary inputs to the previously simulated 8-bit multipliers. The final resultant output of the 16-bit multiplier P (0-31) occurred after the operation.

Using the circuitry in Figure 5(a), a 16-bit multiplier is designed calling the previously designed 8-bit multiplier which is equipped with the proposed compressors along with the exact compressor too.

By simulating and executing this 16-bit multiplier a deeper knowledge regarding the VLSI constraints occurs which is Power consumption, speed of operation, accuracy and area on chip. By this comparative evaluation of the resultant values, we can select the compressor to be used according to the application of its need and thereby benefiting the circuitry. Using the Verilog coding the multiplier is designed and by using the cadence tools the netlist file is generated and the power, area, timing values are simulated and the final physical design of the multiplier equipped with the compressors is produced and can be used according to the choice of the user.

*32-bit multiplier generated using 16-bit multipliers:*

Equipping the proposed compressors, an 8-bit multiplier has been designed and their respective outcomes have been understood. In order to design 16-bit multiplier and 32-bit multiplier, a calling circuitry has been used so that the previously designed 8-bit multiplier can be used according to the necessity and the result can be achieved.

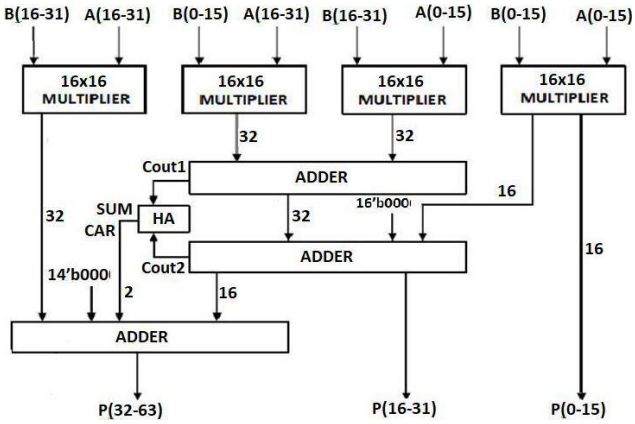


**Figure 5(b): 32-bit multiplier design using 16-bit multipliers**

Here the ADDER block represents a regular 32-bit concatenation adder to sum up the given two 32-bit inputs which are the outputs of the previous stage of the 16-bit multipliers. The A (0-7), A (8-15) and B (0-7), B (8-15) are the binary inputs to the previously simulated 8-bit multipliers. The final resultant output of the 32-bit multiplier P (0-63) occurred after the operation.

The concatenated input 14'b000… meant 14'b00000000000000 And 16'b000… meant 16'b0000000000000000

Using the circuitry in Figure 5(b), a 32-bit multiplier is designed calling the previously designed 16-bit multiplier which is equipped with the proposed compressors along with the exact compressor and evaluated

**STUDY OF ACCURACY OF THE MULTIPLIERS REALIZED BY THE PROPOSED COMPRESSORS**

The output quality of the multipliers designed by incorporating with the proposed dual quality 4:2 compressors can be determined by using a comparative

evaluation of the parameters such as delay, area, power and accuracies. In addition to these there are many comparison factors such as mean and others which can be used. EDA or EDP products of various outputs can also be used to define their strength and make them possible to be adapted in various solicitations. EDP alone can be used as it is considered to be a major factor to choose the desired device for the operations of its use.

The error percentage of the outputs can be calculated using certain error metrics:

Error Percentage = [(Actual output- Resultant output ) ÷ Actual output] * 100 %

EDP = (Energy * Delay)

EDA = (Energy * Delay * Area)

Using all these parameters the multiplier results are compared and ranked accordingly to determine their characteristics. For solicitations where low power is the main criteria one of these can be selected based on their rankings. Similarly based on their interests and criteria of priority the multipliers are opted under use.

As the fallouts showcase, the raise in the blunder constraints would conclude in further generalization (nominalization) of the entity. This could replicate enhanced design bounds of use. This leaning possibly won't monitor the regulation once the build of the product (×)-block alters its form. Especially for most of the constraints, the grades of the anticipated multipliers in this rag are enhanced than those of the multipliers previously existed in-[14] and [15]. A cadence tool with 45nm technology has been employed to extract the evaluation results.

## VI. EVALUATED RESULTS

In this fragment, the efficiencies of the put-forward 4:2compressors in the approximate mode and the Exact 4:2 compressors are evaluated using Verilog coding equipping them in the multiplier and the simulated results are gathered. The outputs are simulated according to the given inputs. By studying the results and comparing them to the conventional/ theoretical multiplier outputs, the error percentage of the multiplier is calculated.

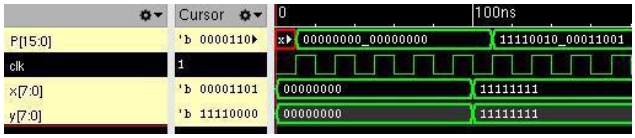


**Figure 7(a). Binary output of DQ4:2C1 used 8-bit Multiplier**


**Figure 7(b). Decimal output- DQ4:2C1 used 8-bit Multiplier**

Let us consider from Figure 6(b) the 2nd input and output of the approximation, the error percentage can be calculated as: Input = 255*255 => Actual Output = 65025

Resultant output = 61977

Error percentage = [(65025-61977) ÷ 65025] * 100%

= [3048 ÷ 65025] *

100 =4.6874 %

Figure 7(a) is its binary output. Further the characteristics of each multiplier that are designed equipping the proposed compressors are simulated using cadence tools in order to study their characteristics such as power, area, delay and accuracy values respectively.

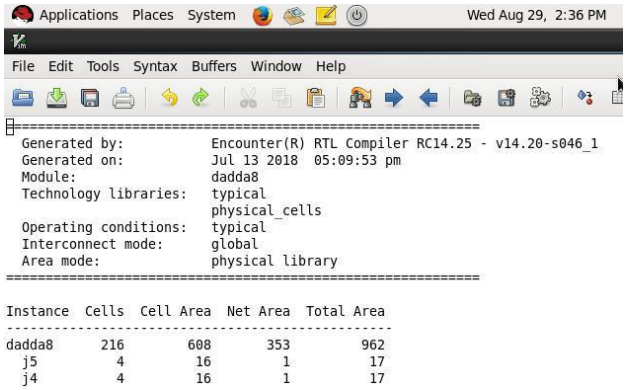


**Figure 7(c). Area report log file simulated using Cadence tool**

Here the area log file has been reported in Figure 7(c). In the similar manner all the log files of Power, area and delay are generated after the simulation using which the accuracy is also calculated. Finally the Physical Design of the multiplier is also simulated as shown in the below Figure 7(d).

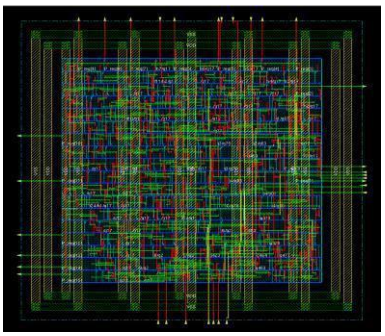

**Figure 7(d): Physical Design of 8-bit multiplier**

*Comparative Evaluation of Design parameters of the proposed compressors utilized multipliers are as shown:*

*The* TABLE: 1, TABLE: 2 & TABLE: 3 *are for the 8-bit, 16-bit & 32-bit Multipliers respectively.*

**TABLE: 1**

| 8-Bit MULTIPLIER | | | | | | | |
|---|---|---|---|---|---|---|---|
| | FAHA | CONVENTIONAL | EXACT | DQ4:2C1 | DQ4:2C2 | DQ4:2C3 | DQ4:2C4 |
| DELAY(pS) | 421 | 1 | 1 | 28 | 1 | 58 | 0 |
| POWER(nW) | 415763 | 2615345 | 2507427 | 2496195 | 2463565 | 2008712 | 2151167 |
| Total AREA($\mu m^2$) | 748 | 1060 | 921 | 962 | 997 | 948 | 1213 |
| ENERGY(fJ) | 175036223 | 2615345 | 2507427 | 69893460 | 2463565 | 116505296 | 2151167 |
| EDP(pSXfJ) | $7.3690\times10^{10}$ | 2615345 | 2507427 | $0.19570\times10^{10}$ | 2463565 | $0.675730\times10^{10}$ | 2151167 |

**TABLE: 2**

| 16-Bit MULTIPLIER | | | | | | | |
|---|---|---|---|---|---|---|---|
| | FAHA | CONVENTIONAL | EXACT | DQ4:2C1 | DQ4:2C2 | DQ4:2C3 | DQ4:2C4 |
| DELAY(pS) | 741 | 0 | 0 | 1 | 1 | 0 | 0 |
| POWER(nW) | 6369510 | 8931670 | 8533548 | 8217146 | 8342364 | 8182454 | 8498758 |
| Total AREA($\mu m^2$) | 3570 | 5490 | 4570 | 4709 | 4838 | 5659 | 6073 |
| ENERGY(fJ) | $4.71980\times10^{10}$ | 8931670 | 8533548 | 8217146 | 8342364 | 8182454 | 8498758 |
| EDP(pSXfJ) | $3497X10^{10}$ | 8931670 | 8533548 | 8217146 | 8342364 | 8182454 | 8498758 |

**TABLE: 3**

| 32-Bit MULTIPLIER | | | | | | | |
|---|---|---|---|---|---|---|---|
| | FAHA | CONVENTIONAL | EXACT | DQ4:2C1 | DQ4:2C2 | DQ4:2C3 | DQ4:2C4 |
| DELAY(pS) | 1375 | 0 | 0 | 0 | 0 | 0 | 0 |
| POWER(nW) | 22985589 | 27793665 | 26118200 | 25928066 | 26258460 | 24944754 | 26543264 |
| Total AREA($\mu m^2$) | 20233 | 23890 | 20650 | 21234 | 21629 | 22349 | 25631 |
| ENERGY(fJ) | $3.1605184\times10^{10}$ | 27793665 | 26118200 | 25928066 | 26258460 | 24944754 | 26543264 |
| EDP(pSXfJ) | $4345\times10^{10}$ | 27793665 | 26118200 | 25928066 | 26258460 | 24944754 | 26543264 |

*Accuracy analysis*

Considering all the yields of the multiplier, at the approximate range the accuracy metrics of 8-bit, 16-bit, 32-bit multipliers are analyzed. The accuracies of8-bit, 16-bit and 32-bit Dadda multiplier utilizing DQ 4:2 compressors for their terms reduction proposed in [28] are too low which varied from 38% to 68% where in this paper due to the developed structure of the dadda reduction, the accuracies improved to a very great extent almost up to 70% to 90%.

*MAC Unit customization:*

A MAC Unit has been designed as in Figure 7(e) integrating the Multiplier utilized portion with our proposed EXACT compressor utilized Multiplier resulting in a successful yield of customized MAC unit, As a MAC unit plays a crucial role in much major application, its customization provides with very good circuitry advancement. It is used in many DSP solicitations [26], Microprocessor and many other arithmetic solicitations.

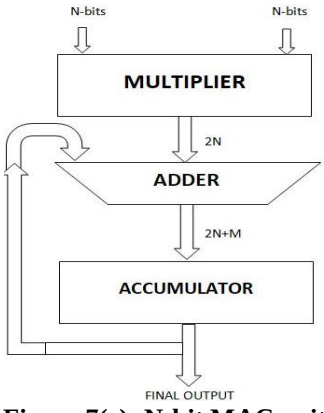


**Figure 7(e): N-bit MAC unit**

## VII. CONCLUSION

In this work, one of the most used arithmetic block that supports crucial operations in many circuitry i.e. the Multiplier is customized according to the requirement based on Area, Delay, Power consumption and Accuracy by using

Quality 4:2 Compressors in the reduction stages as an alternative to its basic Adders where ever they can be used. The usage of these compressors made the multiplier suitable to be used for several solicitations which are in need of the desired characteristics such as low power consumption, decreased area on chip, lower Delay time i.e. improved speed of operation at the cost of and at the cost of reasonable accuracy. Solicitations such as Multimedia processing, Machine learning doesn't require that precision or accuracy in the results at all times, but do require lower power consumption and higher rate of speed. This is achieved by using the proposed quality 4:2compressors in the reduction of the layers of Multipliers. One of the proposed compressors, the EXACT one doesn't even compromise in terms of accuracy and maintains such higher rates of speed and lowered power consumption which is highly desired characteristics in case of all the solicitations that require the presence of multipliers. The proposed approximate compressors have been utilized in 8-bit, 16-bit and 32-bit multipliers using Verilog coding and the consumptions have been studied. These synthesized results reveal that the proposed ones are highly power efficient. It is also proven that these have better precision than the previously designed approximate parts in [28]. The resultant values of power, area, and speed are studied and a comparative evaluation using various parameters and factors is done, providing the users to choose their range of multipliers equipped with compressors based on their requirements and rankings. The multipliers are utilized and an advanced 16-bit and 32-bit MAC unit has been designed to assess them, which is yielding a better result when compared to the existing units of operation while using the exact one, and while using the approximate ones the results are efficient by power consumption or area or delay reduction which concludes that the proposed compressor resulted in a good yield of outcomes supporting the requirements of the VLSI design metrics and solicitations.

## VIII. ACKNOWLEDGMENT

The authors would like to gratefully acknowledge the training & technical support of the CITD- Central Institute of Tool Design, Hyderabad, Telangana along with the guidance of Andhra University College of Engineering (AUCE (A)), Visakhapatnam, Andhra Pradesh.

The authors put-forward that this is Self-Financed.

## REFERENCES

1. P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in *Proc. 24th Int. Conf. VLSI Design*, Jan. 2011, pp. 346–351.
2. D. Baran, M. Aktan, and V. G. Oklobdzija, "Multiplier structures for low power applications in deep-CMOS," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 1061–1064.
3. S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage scalable high-speed robust hybrid arithmetic units using adaptive clock-ing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 9, PP1301–1309, Sep. 2010.
4. O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," *IEEE Trans. Circuits Syst. II, Express Briefs*, doi: 10.1109/TCSII.2016.2633307.
5. A. Sampson *et al.*, "EnerJ: Approximate data types for safe and general low-power computation," in *Proc. 32nd ACM SIGPLAN Conf. Program. Lang. Design Implement. (PLDI)*, 2011, pp. 164–174.
6. A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 846–857, May 2015.
7. J. Joven *et al.*, "QoS-driven reconfigurable parallel computing for NoC-based clustered MPSoCs," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1613–1624, Aug. 2013.
8. R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2013, pp. 48–54.
9. M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.
10. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal process-ing and classification applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 6, pp. 1180–1184, Jun. 2015.
11. S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbi-ased multiplier for approximate applications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, Nov. 2015, pp. 418–425.
12. K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-power high-speed mul-tiplier for error-tolerant application," in *Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC)*, Dec. 2010, pp. 1–4.
13. H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 4, pp. 850–862, Apr. 2010.
14. A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.
15. C. H. Lin and I. C. Lin, "High accuracy approximate multiplier with error correction," in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*, Oct. 2013, pp. 33–38.
16. C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *Proc. Conf. Design, Autom. Test Eur. (DATE)*, 2014, Art. no. 95.
17. R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram, "RoBA multiplier: A rounding-based approximate mul-tiplier for high-speed yet energy-efficient digital signal process-ing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, doi: 10.1109/TVLSI.2016.2587696.
18. C. H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low- power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 10, pp. 1985–1997, Oct. 2004.

19. D. Baran, M. Aktan, and V. G. Oklobdzija, "Energy efficient imple-mentation of parallel CMOS multipliers with improved compressors," in *Proc. ACM/IEEE Int. Symp. Low-Power Electron. Design (ISLPED)*, Aug. 2010, pp. 147–152.
20. J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1760–1771, Sep. 2013.
21. (2016). *NanGate—The Standard Cell Library Optimization Company*. [Online]. Available: http://www.nangate.com/
22. M. S. K. Lau, K. V. Ling, and Y. C. Chu, "Energy-aware probabilistic multiplier: Design and analysis," in *Proc. Int. Conf. Compil., Archit., Synth. Embedded Syst.*, 2009, pp. 281–290.
23. New approximate multiplier for low power digital signal processing, by Farzad Farshchi, Sied Mehdi Fakhrale, https://www.researchgate.net/conference-event/CADS_International-Symposium-on-Computer-Architecture-Digital-Systems_2013/4520A
24. Design Technique for Faster Dadda Multiplier, B. Ramkumar, V. Sreedeep and Harish M Kittur, Member, IEEE.
25. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
26. Trace.eas.asu.edu. (2016). *YUV Sequences*. [Online]. Available: http://trace.eas.asu.edu/yuv/
27. A Proposed Wallace Tree Multiplier Using Full Adder and Half Adder, International Journal of Innovative Research in Electrical, electronics, instrumentation and control Engineering, Vol. 4, Issue 5, May 2016, Swathi AC, Yuvaraj.T
28. Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram, IEEE transactions on very large scale integration (vlsi) systems, vol. 25, no. 4, April 2017.
29. Design and performance analysis of Multiply-Accumulate (MAC) unit, 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], IEEE explorer, 05 March 2015
30. Multiplier-Accumulator (MAC) Unit, International Journal of Digital Application & Contemporary Research Website: www.ijdacr.com (Volume 5, Issue 3, October 2016).
31. Design and Implementation of High Performance 4-bit Dadda Multiplier using Compressor Rathisha Shetty, Mr. Mahesh,B.Neelagar.